View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Lirias

# Combining Logic and Business Rule Systems

Pieter Van Hertum
*pieter.vanhertum@cs.kuleuven.be*

Department of Computer Science, KU Leuven

## 1 Introduction and Motivation

To model complex knowledge, for solving knowledge intensive problems, one needs formalisms and systems that can represent this knowledge in a natural, compact and modular way. Business Rule Systems (BRS) are a widespread way of modelling knowledge intensive problems, with many applications in, among others, planning, supply chain management and expert systems. These systems model knowledge in the form of "if-then" rules "**if** body **then** head". This formalism is used in academic as well as industrial context, mostly because of its readability and the ease with which one can modify the behaviour of rules. Rich interface tools are available, like for example IBM's JRules and Drools. A second advantage of this formalism is the ease to reason with the rules in an automated way for which efficient algorithms exist.

From our point of view, the rigid rule based formulation of the knowledge has some major disadvantages. The knowledge that is modelled in a system, is strictly packed in these procedural style rules. These rules are equipped with a procedural style semantics. As a result, the only method of inference on these rules is execution (given information that occurs in the body of some of the rules, infer new information occurring in the heads of these rules). These limitations are not just an issue of the specific Business Rule System, but of the underlying rules and the ambiguity in their semantics. Take these two rules:

- **if** $Raining$ **then** $Wet(Car)$
- **if** $Age(Person1) > Age(Person2)$ **then** $Older(Person1, Person2)$

These two rules behave the same semantically, as long as forward reasoning (execution) is used. If it is raining, the car will be wet and if the age of the first person is larger than the age of the second one, we will call the first one older. We notice however that the intended semantics can be different if the knowledge is used in a different direction. Say that *Ann* is older than *Bob*, it can be derived that the age of *Ann* is a larger number than the age of *Bob*. If the car is wet on the other hand, it doesn't have to rain.

Alternatively, formalisms based on first order logic could be used to model business logic. They result in an increase in expressive power, but one might be concerned about the impact on the efficiency since an increase in expressivity can imply an increase in complexity. However, a lot of improvements on these computational tools have been made since these formalisms were developed.

Where once it was thought it was impossible to solve real life problems with first order logic, research now shows otherwise [1].

In what follows we propose two new ways of reasoning in the context of BRS that are going to be investigated in this research. The first new way of reasoning, is called **Exploration of specification**. When developing a rule set for a Business Rule System, an important part is testing the system for correctness. To do this properly, a user should be able to ask questions like "Given partial information X, is result Y ever possible?".

Another inference we want to study is **Generation of explanations**. A rule system is in many contexts used for decision problems. In many cases, it is very interesting for a user to know why a certain decision is made. This is not only useful for debugging purposes, this can also be a rich feature in the day to day use of such a system.

Another aspect of this research is solving the problems a rule system has when reasoning with incomplete information. In many real life applications, large amounts of data can be relevant to the problem at hand. However, sometimes, a small fraction of that data can be enough to take a decision. In the current state of the art, a business rule system is not capable of reasoning with incomplete information in a general way.

## 2 Completed and Future Work

In our research group, the reasoning tool IDP is developed, based on $\mathrm{FO}(\cdot)$, a set of conservative extensions of first order logic [2]. We believe this tool is perfectly suited for addressing the issues mentioned above since (1) the $\mathrm{FO}(\cdot)$ logic is specifically developed for handling incomplete knowledge ; (2) the IDP system can be easily extended with new inferences and language constructs; (3) multiple concepts and constructs that are needed are already incorporated in the IDP system.

A feasibility study of modelling Business Rule knowledge in IDP has been done. In [3] it is also studied what features the IDP system has that extend the current BR systems and what features are still missing in the IDP system to correctly formalize BR knowledge. The next stage of the research, the development of the new language constructs from the ICLP publication is followed by the development of the new inferences, discussed above.

## References


1. Bruynooghe, M., Blockeel, H., Bart, B., De Cat, B.: Predicate logic as a modeling language: Modeling and solving some machine learning and data mining problems with idp3. CoRR abs/1309.6883 (2013)
2. De Cat, B., Bogaerts, B., Bruynooghe, M., Denecker, M.: Predicate logic as a modelling language: The IDP system. CoRR abs/1401.6312 (2014)
3. Van Hertum, P., Vennekens, J., Bogaerts, B., Devriendt, J., Denecker, M.: The effects of buying a new car: an extension of the IDP knowledge base system. TPLP 13(4-5-Online-Supplement) (2013)